(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10333905 A

(43) Date of publication of application: 18 . 12 . 98

(51) Int. Cl G06F 9/38
G06F 9/38

(21) Application number: 09141687

(22) Date of filing: 30 . 05 . 97

(71) Applicant: KOFU NIPPON DENKI KK

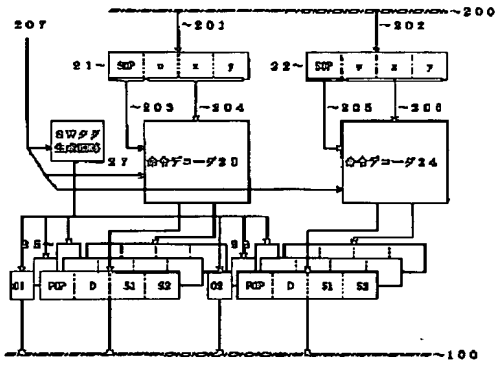
(72) Inventor: TAKATO MASAHIKO

(54) INFORMATION PROCESSOR FOR SUPER SCALAR SYSTEM

(57) Abstract:

PROBLEM TO BE SOLVED: To correctly control even the occurrence of even to be processed between software instructions by imparting a tag to a software instruction unit and recognizing the break of the software instruction.

SOLUTION: As for the software instructions stored in a software instruction queue, by an instruction issuance permission instruction 207, a first SW instruction is decoded by an instruction decoder 23, a second SW instruction is decoded by the instruction decoder 24 and they are developed to plural firmware instructions. At the time, the first SW instruction is developed to the two pieces of firmwares, the second SW instruction is developed to the three pieces of the firmwares and they are stored in firmware instruction queues 25 and 26. Further, an SW tag number 01 and the two firmware instructions are stored in the firmware instruction queue 25, the SW instruction tag number 02 and the three firmware instructions are stored in the firmware instruction queue 26 and the break of the software instruction is recognized.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－333905

(43)公開日　平成10年(1998)12月18日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/38	３１０	Ｇ０６Ｆ　9/38	３１０Ｆ
	３７０		３７０Ａ

審査請求　有　　請求項の数３　ＯＬ　（全　６　頁）

(21)出願番号　特願平9－141687

(22)出願日　平成９年(1997)５月30日

(71)出願人　000168285
甲府日本電気株式会社
山梨県甲府市大津町1088－３

(72)発明者　高遠　雅彦
山梨県甲府市大津町1088－３　甲府日本電気株式会社内

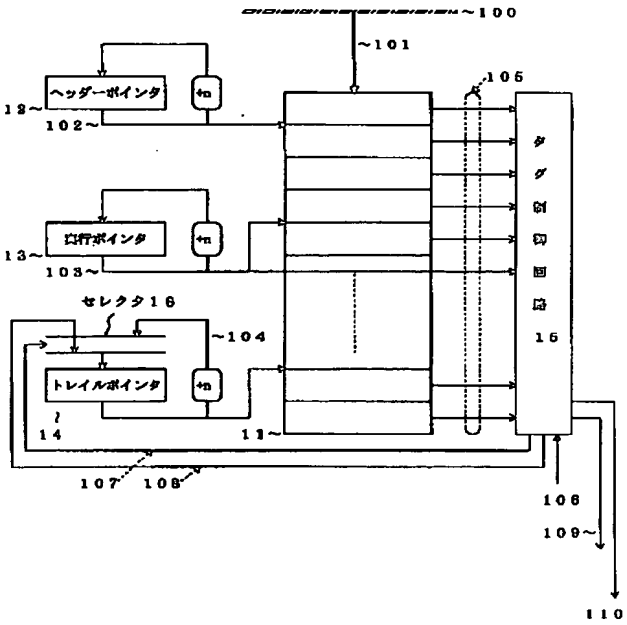
(74)代理人　弁理士　山下　穣平

(54)【発明の名称】　スーパースカラー方式の情報処理装置

(57)【要約】

【課題】　ソフトウェア命令間の区切りを認識でき、ソフトウェア命令間で処理しなければならないイベントが発生した場合でも正しく制御することを可能としたスーパースカラー方式の情報処理装置を提供する。

【解決手段】　ソフトウェア命令を複数のファームウェア命令に展開してプロセッサの１クロックサイクル中に前記複数個のファームウェア命令を実行することが可能な情報処理装置において、前記ソフトウェア命令単位にタグ１０５を付与することで前記ソフトウェア命令の区切りを認識する手段１５を具備したことを特徴とするスーパースカラー方式の情報処理装置。



1

【特許請求の範囲】

【請求項１】　ソフトウェア命令を複数のファームウェア命令に展開してプロセッサの１クロックサイクル中に前記複数個のファームウェア命令を実行することが可能な情報処理装置において、

前記ソフトウェア命令単位にタグを付与することで前記ソフトウェア命令の区切りを認識する手段を具備したことを特徴とするスーパースカラー方式の情報処理装置。

【請求項２】　ソフトウェア命令を複数のファームウェア命令に展開してプロセッサの１クロックサイクル中に複数個のファームウェア命令を実行することが可能な演算処理装置において、

前記ソフトウェア命令を複数のファームウェア命令に展開する際に、前記ソフトウェア命令毎に識別可能なＳＷタグを付与する回路２７と、

前記ファームウェア命令毎に必要な登録内容とＳＷタグをｉｎｏｒｄｅｒに登録するリオーダバッファ１１と、

前記リオーダバッファ１１に登録したエントリの上限を示すヘッダーポインタ１２と、

現在実行中の命令が格納されているエントリを示す実行ポインタ１３と、

前記エントリの下限を示すトレイルポインタ１４と、

前記ソフトウェア命令間のイベント処理が発生した場合に、前記ソフトウェア命令の区切りを判別し、前記トレイルポインタを前記ソフトウェア命令の区切りに移動するタグ制御回路１５と、を有することを特徴とするスーパースカラー方式の情報処理装置。

【請求項３】　前記タグ制御回路１５からタグ情報１１０とキャンセル指示１０９を受け取り、各ステージ毎に持ち回っているＳＷタグ番号３１，３２と比較するタグチェック回路３３，３４を有することを特徴とする請求項２記載のスーパースカラー方式の情報処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複雑なソフトウェア命令を単純なファームウェア命令に展開し、プログラム中に書かれている命令列とは異なる順番で実行する情報処理装置に関し、特に、スーパースカラー方式の情報処理装置に関する。

【０００２】

【従来の技術】従来、演算処理装置においては性能を上げるために複数の命令をオーバラップして実行するパイプライン処理が行われている。

【０００３】データ依存のために、ある命令が実行できない場合でも、その命令よりも後の命令で実行できる命令があれば、その命令を先に実行することによってデータ依存による性能低下を避ける方法がある。この方法はプログラム中の命令列とは異なる順番で命令を実行する方法でｏｕｔ－ｏｆ－ｏｒｄｅｒ実行と呼ばれる。

【０００４】ｏｕｔ－ｏｆ－ｏｒｄｅｒで実行した命令

2

の結果をそのままレジスタファイルに格納すると、例外が発生した場合に問題となる。例外が発生するとその例外のために完了できなかった命令のアドレスが退避され、例外の処理を行った後、退避された命令のアドレスから再び実行が開始される。

【０００５】ところが、ｏｕｔ－ｏｆ－ｏｒｄｅｒ実行であるために、退避された命令アドレスの後続の命令が例外発生時にすでに実行を完了していたとすると、その命令は例外処理のあとに再び実行されてしまう。この問題を解決するためにリオーダバッファが提案されている。

【０００６】特開平６－１９７０７号公報の（１３）頁に記載されている従来のリオーダバッファのブロック図を用いて説明する。図７は、この従来のリオーダバッファのブロック図であり、図７において、１２１はリオーダバッファであり、データを格納するデータフィールド５６とその格納先のレジスタ番号（デスティネーションレジスタ番号）を格納するデスティネーションレジスタフィールド５５からなるエントリー１２２を複数持ち、ｏｕｔ－ｏｆ－ｏｒｄｅｒで得られた演算器５２の結果データを一旦格納し、これをプログラムに書かれている命令列の順番でレジスタファイル５１にデータを格納するものである。

【０００７】リオーダバッファ１２１に保持されたデータをレジスタファイル５１に格納する前に、後続の命令がそのデータを必要とする場合、レジスタファイル５１からの読み出しを禁止し、リオーダバッファ１２１からデータを供給する必要がある。これは、リオーダバッファ１２１の各エントリー１２２に依存検出回路５７を設け、エントリー１２２のデスティネーションレジスタフィールド５５に格納されているデスティネーションレジスタ番号と後続命令のソースレジスタ番号６０及び６１を比較することによってデータ依存が検出され、データ依存情報６４，６５によってエントリー１２２のデータフィールド５６の出力が制御される。

【０００８】ここで、リオーダバッファ１２１に同じデスティネーションレジスタ番号を持つエントリーが複数ある場合、最新データを選択しなければならない。そのため、通常、比較した結果、得られる複数の一致信号の中から優先度をつけて選ぶ回路を有している。

【０００９】

【発明が解決しようとする課題】第１の問題点は、従来の技術において、複雑なソフトウェア命令を単純なファームウェア命令に展開し、プログラム中に書かれている命令列とは異なる順番で実行する演算処理装置において、ソフトウェア命令実行中にイベント事象が発生した場合、ソフトウェア命令間でイベント処理を行うことができないことである。

【００１０】その理由は、ソフトウェア命令を複数のファームウェア命令に展開し、命令の実行順序もｏｕｔ－

ｏｆ－ｏｒｄｅｒであるためソフトウェア命令単位の区切りが判別できないからである。

【００１１】［発明の目的］本発明の目的は、ソフトウェア命令単位のタグビットをリオーダバッファに登録することにより、ソフトウェア命令間の区切りを認識でき、ソフトウェア命令間で処理しなければならないイベントが発生した場合でも正しく制御することを可能とした回路を提供することにある。

【００１２】

【課題を解決するための手段】本発明は、前述した課題を解決するための手段として、ソフトウェア命令を複数のファームウェア命令に展開してプロセッサの１クロックサイクル中に前記複数個のファームウェア命令を実行することが可能な情報処理装置において、前記ソフトウェア命令単位にタグを付与することで前記ソフトウェア命令の区切りを認識する手段を具備したことを特徴とするスーパースカラー方式の情報処理装置を提供するものである。

【００１３】また、ソフトウェア命令を複数のファームウェア命令に展開してプロセッサの１クロックサイクル中に複数個のファームウェア命令を実行することが可能な演算処理装置において、前記ソフトウェア命令を複数のファームウェア命令に展開する際に、前記ソフトウェア命令毎に識別可能なＳＷタグを付与する回路２７と、前記ファームウェア命令毎に必要な登録内容とＳＷタグをｉｎｏｒｄｅｒに登録するリオーダバッファ１１と、前記リオーダバッファ１１に登録したエントリの上限を示すヘッダーポインタ１２と、現在実行中の命令が格納されているエントリを示す実行ポインタ１３と、前記エントリの下限を示すトレイルポインタ１４と、前記ソフトウェア命令間のイベント処理が発生した場合に、前記ソフトウェア命令の区切りを判別し、前記トレイルポインタを前記ソフトウェア命令の区切りに移動するタグ制御回路１５と、を有することを特徴とするスーパースカラー方式の情報処理装置を、上記手段とするものである。

【００１４】また、前記タグ制御回路１５からタグ情報１１０とキャンセル指示１０９を受け取り、各ステージ毎に持ち回っているＳＷタグ番号３１，３２と比較するタグチェック回路３３，３４を有することを特徴とするスーパースカラー方式の情報処理装置でもある。本発明のスーパースカラー方式は、イベント事象が発生した場合にソフトウェア命令間を認識し、ソフトウェア命令間で正しくイベント処理が行える回路を有することが特徴である。

【００１５】具体的には、ソフトウェア命令毎に識別可能なＳＷタグを付与する回路２７と、ファームウェア命令毎に必要な登録内容とＳＷタグ番号を登録するリオーダバッファ１１と、リオーダバッファ１１のエントリーの上限を示すヘッダーポインタ１２と、現在実行中の命令が格納されているエントリーを示す実行ポインタ１３と、エントリーの下限を示すトレイルポインタ１４と、ソフトウェア命令の区切りを判別するタグ制御回路１５を具備することを特徴とする。

【００１６】

【作用】本発明によれば、ソフトウェア命令単位にＳＷタグ番号を付与し、ファームウェア命令の情報とともにＳＷタグ番号もリオーダバッファに登録することで、現在実行中のソフトウェア命令を判断することができる。

【００１７】これにより、イベント事象が発生した場合でもリオーダバッファからＳＷタグ番号を抽出することで、ソフトウェア命令間を正しく判断することができるため、イベント処理を正常に行うことができる。

【００１８】また、ＳＷタグ番号を各ステージの情報と合わせて持ち回っており、タグ制御回路からキャンセル対象のＳＷタグを各ステージに通知し、各ステージでキャンセル対象のステージか否かを判断することで無効なデータのみをキャンセルすることができる。

【００１９】

【実施例】次に、本発明の実施例について図面を参照して説明する。

【００２０】［構成］図１は、本発明のリオーダバッファの外部構成を示したブロック図である。

【００２１】図１を参照すると、本発明の実施例は、ファームウェア情報、ＳＷタグを登録するリオーダバッファ１１と、リオーダバッファに登録されている最古のファームウェア命令を指すヘッダーポインタ１２と、現在実行中のファームウェア命令を指す実行ポインタ１３と、リオーダバッファに次に登録する番地を指すトレイルポインタ１４と、リオーダバッファ１１に登録されているＳＷタグ情報１０５とイベント通知１０６からトレイルポインタの制御情報を生成するタグ制御回路１５から構成される。

【００２２】図２は、ソフトウェア命令をファームウェア命令に展開するブロック図である。

【００２３】ソフトウェア命令を格納するソフトウェア命令キュー２１，２２とソフトウェア命令のＯＰコードと各フィールド情報からファームウェア命令を生成する命令デコーダ２３，２４と、命令デコーダ２３，２４から生成されたファームウェア命令を格納するファームウェア命令キュー２５，２６と、実行部からの発行許可指示によりＳＷ命令単位にタグを生成するタグ生成回路２７から構成される。

【００２４】図３は、各パイプラインキャンセル回路を示すブロック図である。

【００２５】各ステージ毎にＳＷタグ番号３１，３２を格納する手段と、タグ比較を行うタグチェック回路３３，３４と、比較結果でキャンセル対象であることを認識した場合に、後述するバリッド３５，３６をクリアする論理積３７，３８から構成される。

5

【００２６】［動作］次に、本発明の実施例の動作について、図２を参照して詳細に説明する。

【００２７】ソフトウェア命令をファームウェア命令に展開するイメージを図４に示す。図４に示すように、１つのソフトウェア命令は、１～ｎ個のファームウェア命令に展開される。図４の例だと、ＳＷ命令Ａは２つのファームウェア命令に展開され、ＳＷ命令Ｂは３つのファームウェア命令に展開される。

【００２８】また、ＳＷ命令タグ番号は、ソフトウェア命令単位に付与されるため、ＳＷ命令Ａから展開されたファームウェア命令は同一のＳＷ命令タグ番号“０１”、ＳＷ命令Ｂから展開されたファームウェア命令は同一のＳＷ命令タグ番号“０２”が付与される。

【００２９】ソフトウェア命令バス２００には、命令フェッチ部から供給された複数のソフトウェア命令が存在する。ソフトウェア命令キュー２１，２２には、次に実行するソフトウェア命令をソフトウェア命令バス２００より順次格納する。図２の場合、キューの数は２個としているが、いくつでもかまわない。

【００３０】図４を例にすると、ＳＷ命令Ａはソフトウェア命令キュー２１に、ＳＷ命令Ｂはソフトウェア命令キュー２２に格納される。

【００３１】ソフトウェア命令キューに格納されているソフトウェア命令は、命令発行許可指示２０７によって、ＳＷ命令Ａは、命令デコーダ２３に、ＳＷ命令Ｂは命令デコーダ２４によりデコードされ、複数のファームウェア命令に展開される。この場合、ＳＷ命令Ａは２個の、ＳＷ命令Ｂは３個のファームウェアに展開され、ファームウェア命令キュー２５，２６に格納される。

【００３２】また、命令発行許可指示２０７によりＳＷタグ生成回路２７からソフトウェア命令単位にタグ番号が生成され、ファームウェア命令と合わせて、ファームウェア命令キュー２５，２６に格納される。ＳＷタグ番号はソフトウェア命令毎に昇順に割り振られる。つまり、古い命令のＳＷ命令タグ番号程、値が小さくなる。

【００３３】以上の動作により、ファームウェア命令キュー２５にはＳＷタグ番号“０１”と２個のファームウェア命令が、ファームウェア命令キュー２６にはＳＷ命令タグ番号“０２”と３個のファームウェア命令が格納されている。

【００３４】ファームウェアキュー２５，２６に格納されたファームウェア命令とＳＷタグ番号はファームウェア命令バス１００に順次送出される。

【００３５】次に、図１を参照して詳細に説明する。ファームウェア命令バス１００に送出されたファームウェア命令の情報とＳＷタグ番号はリオーダバッファ登録パス１０１を経由してリオーダバッファ１１に登録される。

【００３６】リオーダバッファ１１に登録される内容の例を図５に示す。図５に示すように、リオーダバッファ

6

１１には登録内容が有効であるか無効であるかを表すバリッドＶと、ファームウェア命令で指定されているデスティネーションレジスタ番号と、ペンディング状態であるかを判定するビジーフラグＢと、ＳＷ命令タグ番号と、ライトデータ、ステータス情報などが含まれている。これらの情報がファームウェア命令単位に各エントリーに登録される。

【００３７】リオーダバッファ１１は、循環型ＦＩＦＯ構造になっており、各ポインタによって制御される。ヘッダーポインタ１２は、リオーダバッファ１１に登録されている最古のエントリー番号を示し、トレイルポインタ１４は、ファームウェア命令バス１００から次に登録するエントリー番号を示し、実行ポインタ１３は現在実行中であるファームウェア命令のエントリー番号を示している。

【００３８】図６に、図４のファームウェアがリオーダバッファ１１に登録されているイメージ図を示す。各ファームウェア命令の情報はエントリー番号２０～２４に登録されており、ＳＷ命令Ａの１つめファームウェア命令が既に実行が完了しているとする。この場合、エントリー番号２０のステータスには終了（Ｄｏｎｅ）が表示される。その他の命令はまだ実行が完了していないためビジーフラグＢは“１”であり、ステータスは実行中（ｅｘｅｃ）、実行待ち（Ｒｅａｄｙ）、ペンディング状態（Ｗａｉｔ）となっている。この状態では、トレイルポインタ１４は“２５”、実行ポインタ１３は“２１”を指している。

【００３９】この時にイベント通知１０６が通知されると、タグ制御回路１５は現在実行中である命令のエントリー番号を実行ポインタ１３から実行ポインタ番号１０３を経由して受け取る。実行ポインタ１３は“２１”を指しているため、リオーダバッファ１１のエントリー２１に登録してあるＳＷタグ番号“０１”をＳＷタグ情報１０５から受け取る。タグ制御回路１５でエントリー２１のＳＷタグ番号と同一の番号をもつエントリーを検索し、その中で最も新しく登録されたエントリー番号をさらに検索する。

【００４０】検索した結果のエントリー番号（この場合は“２１”）に＋１インクリメントした値をトレイルポインタ番号１０８にセットし、トレイルポインタセット指示１０７を“１”にする。この指示によりセレクタ１６は、トレイルポインタ番号１０８を選択し、トレイルポインタ１４を、現在指しているエントリー番号“２５”から“２２”に変更する。

【００４１】また、タグ制御回路１５はキャンセル指示１０９を発行するとともにキャンセル対象のタグ番号である現在実行中のＳＷタグ番号“０１”に＋１インクリメントした値“０２”をタグ情報１１０に送出する。

【００４２】次に図３を用いてＳＷタグのキャンセル動作を説明する。

7

【００４３】各ステージには、ファームウェア命令を実行するための情報以外に、ＳＷタグ番号３１，３２を一緒に持ち回っており、キャンセル指示１０９とタグ情報１１０を受け取ると自ステージのＳＷタグ番号３１，３２と比較を行い、自ステージのＳＷタグ番号がタグ情報１１０で通知された番号よりも大きければ自ステージの情報は無効であるため、対応するステージのバリッドを論理積３７，３８により“０”に落とす。

【００４４】

【発明の効果】第１の効果は、複雑なソフトウェア命令を単純なファームウェア命令に展開し、プログラム中にかかれている命令列とは異なる順番で実行する演算処理装置において、イベント処理が正しく行える。

【００４５】その理由は、ソフトウェア命令単位にタグを付与することでソフトウェア命令の区切りを認識することができるからである。

【図面の簡単な説明】

【図１】本発明のスーパースカラー回路の一実施例を示すリオーダバッファの外部構成図である。

【図２】本発明のスーパースカラー回路の一実施例を示すソフトウェア命令デコーダ廻りのブロック図である。

【図３】本発明のスーパースカラー回路のパイプラインキャンセル回路を示すブロック図である。

8

＊【図４】本発明のスーパースカラー回路のファームウェア命令の展開イメージ図である。

【図５】本発明のスーパースカラー回路のリオーダバッファの登録内容である。

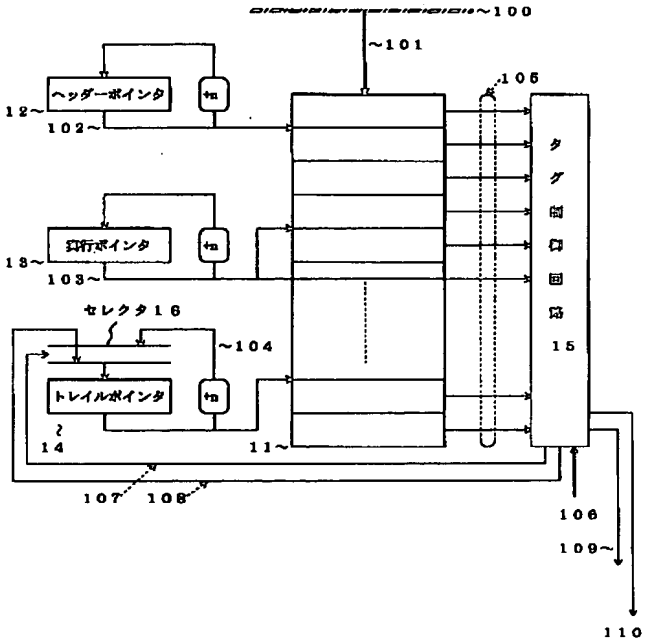
【図６】本発明のスーパースカラー回路の図４のファームウェア命令列のリオーダバッファの登録内容である。

【図７】従来例のリオーダバッファの構成を示すブロック図である。

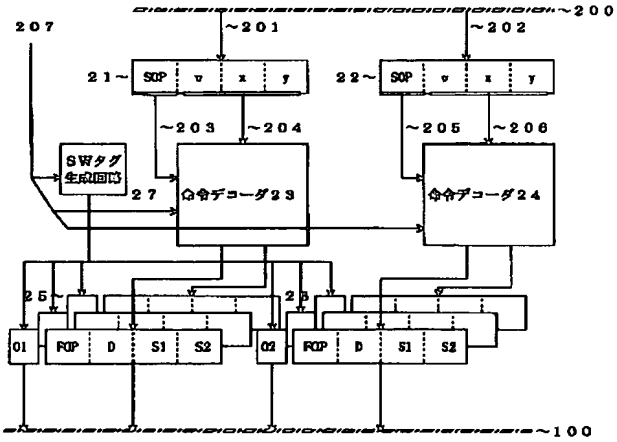
【符号の説明】

１１　リオーダバッファ
１２　ヘッダーポインタ
１３　実行ポインタ
１４　トレイルポインタ
１５　タグ制御回路
２１，２２　ソフトウェア命令キュー
２３，２４　命令デコーダ
２５，２６　ファームウェア命令キューとＳＷタグ番号
２７　ＳＷタグ生成回路
３１，３２　ＳＷタグ番号
３３，３４　タグチェック回路
３５，３６　ステージバリッド
３７，３８　論理積

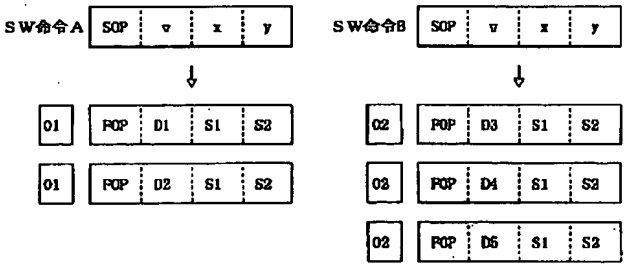
【図１】



【図２】



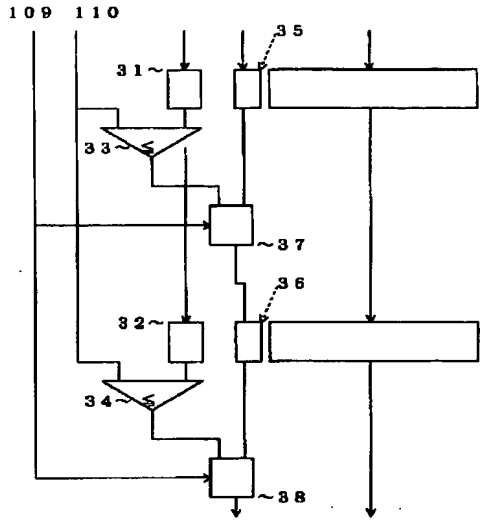
【図４】



【図５】

V	DestReg#	B	SoTag#	Cc	Vr	WriteData	Status

【図３】



【図６】

	V	DestReg#	B	SvtagH	Cc	Tv	WriteData	Status
20#	1	D1	0	01	-	1	01234567	Done
	1	D2	1	01		0		exec
	1	D3	1	02		0		exec
	1	D4	1	02		0		Ready
	1	D5	1	02		0		Wait

【図７】

55 デスティネーションレジスタフィールド
56 データフィールド
57 検出器